

Model-based rate-distortion optimized video-based point cloud compression with differential evolution

Hui Yuan^{1, 2(✉)}, Raouf Hamzaoui¹, Ferrante Neri³, and Shengxiang Yang⁴

¹ School of Engineering and Sustainable Development, De Montfort University, Leicester, UK

² School of Control Science and Engineering, Shandong University, Jinan, China
hui.yuan@dmu.ac.uk, huiyuan@sdu.edu.cn

³ School of Computer Science, University of Nottingham, Nottingham, UK

⁴ School of Computer Science and Informatics, De Montfort University, Leicester, UK

Abstract. The Moving Picture Experts Group (MPEG) video-based point cloud compression (V-PCC) standard encodes a dynamic point cloud by first converting it into one geometry video and one color video and then using a video coder to compress the two video sequences. We first propose analytical models for the distortion and bitrate of the V-PCC reference software, where the models' variables are the quantization step sizes used in the encoding of the geometry and color videos. Unlike previous work, our analytical models are functions of the quantization step sizes of all frames in a group of frames. Then, we use our models and an implementation of the differential evolution algorithm to efficiently minimize the distortion subject to a constraint on the bitrate. Experimental results on six dynamic point clouds show that, compared to the state-of-the-art, our method achieves an encoding with a smaller error to the target bitrate (4.65% vs. 11.94% on average) and a slightly lower rate-distortion performance (on average, the increase in Bjøntegaard delta (BD) distortion is 0.27, and the increase in BD rate is 8.40%).

Keywords: point cloud compression, rate-distortion optimization, rate control, rate and distortion models, differential evolution.

1 Introduction

A static point cloud is a representation of a three-dimensional object, where in addition to the spatial coordinates of a sample of points on the surface of the object, attributes such as color, reflectance, transparency, and normal direction may be used. A dynamic point cloud consists of several successive static point clouds. Each point cloud in the sequence is called a frame. Point clouds are receiving increased attention due to their potential for immersive video experience applications such as virtual reality, augmented reality, and immersive telepresence.

To get a high-quality representation of a three-dimensional object as a point cloud, a huge amount of data is required. To compress point clouds efficiently, the Moving Picture Experts Group (MPEG) launched in January 2017 a call for proposals for point cloud compression technology. As a result, two point cloud compression standards are

being developed: video-based point cloud compression (V-PCC) [1] for point sets with a relatively uniform distribution of points and geometry-based point cloud compression (G-PCC) [2] for more sparse distributions. In this paper, we focus on V-PCC for dynamic point clouds. In V-PCC, the input point cloud is first decomposed into a set of patches, which are independently mapped to a two-dimensional grid of uniform blocks. This mapping is then used to store the geometry and color information as one geometry video and one color video. Next, the generated geometry video and color video are compressed separately with a video coder, e.g., H.265/HEVC [3]. Finally, the geometry and color videos, together with metadata (occupancy map for the two-dimensional grid, auxiliary patch, and block information) are multiplexed to generate the bit stream (Fig. 1 [1]). In the video coding step, compression is achieved with quantization, which is determined by a quantization step size or, equivalently, a quantization parameter (QP).

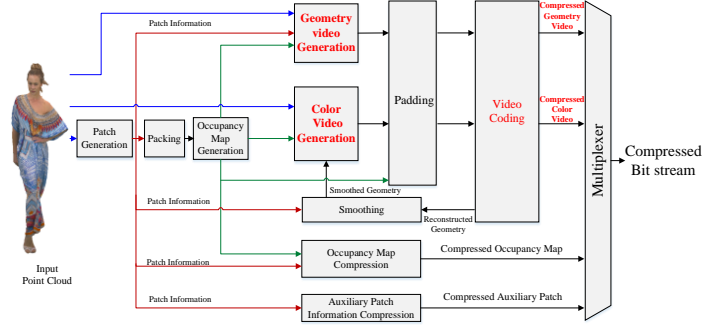


Fig. 1. V-PCC test model encoder [1].

Given a set of M quantization step sizes $\{q_0, \dots, q_{M-1}\}$ and a dynamic point cloud consisting of N frames, an optimal encoding can be obtained by determining for each frame i ($i = 1, \dots, N$) the geometry quantization step size $Q_{g,i} \in \{q_0, \dots, q_{M-1}\}$ and color quantization step size $Q_{c,i} \in \{q_0, \dots, q_{M-1}\}$ that minimize the distortion subject to a constraint R_T on the total number of bits. This can be formulated as the multi-objective optimization problem

$$\min_{\mathbf{Q}_g, \mathbf{Q}_c} [D_g(\mathbf{Q}_g, \mathbf{Q}_c), D_c(\mathbf{Q}_g, \mathbf{Q}_c)] \quad (1)$$

$$s. t. \quad R(\mathbf{Q}_g, \mathbf{Q}_c) = R_g(\mathbf{Q}_g, \mathbf{Q}_c) + R_c(\mathbf{Q}_g, \mathbf{Q}_c) \leq R_T,$$

where $\mathbf{Q}_g = (Q_{g,1}, Q_{g,2}, \dots, Q_{g,N})$, $\mathbf{Q}_c = (Q_{c,1}, Q_{c,2}, \dots, Q_{c,N})$, $D_g(\mathbf{Q}_g, \mathbf{Q}_c)$ is the geometry distortion, $D_c(\mathbf{Q}_g, \mathbf{Q}_c)$ is the color distortion, $R(\mathbf{Q}_g, \mathbf{Q}_c)$ is the total number of bits, $R_g(\mathbf{Q}_g, \mathbf{Q}_c)$ is the number of bits for the geometry information, and $R_c(\mathbf{Q}_g, \mathbf{Q}_c)$ is the number of bits for the color information. Here $D_g(\mathbf{Q}_g, \mathbf{Q}_c) = \frac{1}{N} \sum_{i=1}^N D_{g,i}(\mathbf{Q}_g, \mathbf{Q}_c)$ and $D_c(\mathbf{Q}_g, \mathbf{Q}_c) = \frac{1}{N} \sum_{i=1}^N D_{c,i}(\mathbf{Q}_g, \mathbf{Q}_c)$, where $D_{g,i}(\mathbf{Q}_g, \mathbf{Q}_c)$ and $D_{c,i}(\mathbf{Q}_g, \mathbf{Q}_c)$ are the geometry and color distortions of the i th frame, respectively. Similarly, $R_g(\mathbf{Q}_g, \mathbf{Q}_c) = \sum_{i=1}^N R_{g,i}(\mathbf{Q}_g, \mathbf{Q}_c)$ and $R_c(\mathbf{Q}_g, \mathbf{Q}_c) = \sum_{i=1}^N R_{c,i}(\mathbf{Q}_g, \mathbf{Q}_c)$, where $R_{g,i}(\mathbf{Q}_g, \mathbf{Q}_c)$ and $R_{c,i}(\mathbf{Q}_g, \mathbf{Q}_c)$ are the number of bits for the geometry and color of the i th frame, respectively. In practice, problem (1) is scalarized as follows.

$$\begin{aligned} \min_{\mathbf{Q}_g, \mathbf{Q}_c} [D(\mathbf{Q}_g, \mathbf{Q}_c) = \omega D_c(\mathbf{Q}_g, \mathbf{Q}_c) + (1 - \omega) D_g(\mathbf{Q}_g, \mathbf{Q}_c)] \\ \text{s. t. } R(\mathbf{Q}_g, \mathbf{Q}_c) \leq R_T, \end{aligned} \quad (2)$$

where $\omega \in [0,1]$ is a weighting factor that sets the relative importance of the geometry and color distortions. As the number of possible solutions is M^{2N} , solving the problem with exhaustive search is not feasible when M or N is large as the computation of the distortion and the number of bits requires encoding and decoding the point cloud, which is very time consuming. In this paper, we solve the rate-distortion optimization problem (2) by first developing analytical models for the distortion and bitrate and then applying a metaheuristic based on differential evolution (DE) [4] to the analytical models. There is a need for new models as the existing ones [5,6,7] are not suitable for the rate-distortion optimization problem (2). Note also that the V-PCC standard does not give any solution to problem (2). In the latest MPEG V-PCC test model [8], for example, the QPs for the geometry and color are selected manually: one chooses the QPs of the first frame, and the QP values of the following frames are set according to some fixed rules (e.g., by using the same values for the low delay configuration).

2 Related Work

Only a small number of works [5,6,7] have proposed rate and distortion models for point cloud compression. In [5], the focus is on the point cloud library (PCL) platform [9] for the compression of static point clouds. This platform uses an octree decomposition for geometry compression and JPEG for color compression. Analytical models that describe the relationship between the encoding parameters (the maximum octree level and the JPEG quality factor) and the color distortion D_c and bitrate R are derived with statistical analysis. Let L be the maximum octree level and let J be the JPEG quality factor. The color distortion is modeled as $D_c = sJ^pL^q$, where s, p, q are model parameters. On the other hand, the bitrate is modeled as $\ln R = aLJ + bL + c$, where a, b, c are model parameters. Then, the models are used to formulate the rate-distortion optimization problem as a constrained optimization problem, and an interior point method is applied to solve it. In [6], a similar approach is applied to V-PCC for dynamic point clouds. First, distortion and rate models for the geometry information and color information are derived as follows: $D_g = \alpha_g Q_{g,1} + \delta_g$, $D_c = \alpha_c Q_{c,1} + \beta_c Q_{g,1} + \delta_c$, $R_g = \gamma_g Q_{g,1}^{\theta_g}$, $R_c = \gamma_c Q_{c,1}^{\theta_c}$, where $\alpha_g, \delta_g, \alpha_c, \beta_c, \delta_c, \gamma_g, \theta_g, \gamma_c, \theta_c$ are model parameters. Then, an interior point method is used to minimize the weighted sum of the distortions subject to a constraint on the total number of bits. One limitation of this work is that the distortion and rate models are functions of the quantization steps of the geometry and color information of the first frame only. Thus, these models are only suitable when the quantization steps of the following frames are set according to the default settings of the V-PCC test model and are not appropriate for the general rate-distortion optimization problem (2). In [7], a point cloud is partitioned into seven regions such that the first six regions correspond to the six patches with the largest area in the six projection planes,

and the seventh region consists of all other patches. Then, the geometry and color quantization steps corresponding to each region are optimized separately using the analytical models in [6].

3 Rate and Distortion Models

In this section, we propose new analytical distortion and rate models for V-PCC. For both the geometry distortion and color distortion, we used the symmetric point-to-point distortions based on the mean squared error (MSE) [10]. Moreover, for the color information, we considered only the Y (luminance) component. To compute the actual values of the distortion and bitrate, we used the latest V-PCC test model (TMC2 v12.0) [8], where the encoder settings were modified such that the QPs of the frames can be chosen arbitrarily. Note that TMC2 v12.0 relies on the HEVC Test Model Version 16.20 (HM16.20) [11] to compress the geometry and color videos. In HEVC, the set of QP values is $\{0, \dots, 51\}$, which corresponds to quantization step sizes $\{0.625, \dots, 224\}$. We encoded four frames of the point cloud using the low delay configuration with group of pictures (GOP) structure IPPP.

Table 1. Dependency between the first frame and the second frame for the *basketballplayer* point cloud. Encoding is with the low delay configuration of [8].

$Q_{g,1}$	$Q_{g,2}$	$D_{g,1}$	$D_{g,2}$	$R_{g,1}$	$R_{g,2}$	$Q_{c,1}$	$Q_{c,2}$	$D_{c,1}$	$D_{c,2}$	$R_{c,1}$	$R_{c,2}$
11	11	0.306637	0.320112	126160	97880	11	11	0.000147568	0.000146696	657056	520040
14	11	0.348734	0.316439	103632	101040	14	11	0.00015884	0.0001459	490024	544440
18	11	0.418704	0.323274	92360	108104	18	11	0.000174033	0.00014684	368144	562112
22	11	0.504359	0.334045	74720	113936	22	11	0.000189179	0.000147952	280640	575800
28	11	0.620279	0.309088	64576	119544	28	11	0.000209897	0.000145801	214840	604040
18	18	0.418704	0.434988	92360	70112	18	18	0.000174033	0.000172132	368144	263520
22	18	0.504359	0.446426	74720	72096	22	18	0.000189179	0.000170755	280640	274608
28	18	0.620279	0.447552	64576	76344	28	18	0.000209897	0.000170923	214840	298392
36	18	0.824124	0.442594	53760	81296	36	18	0.00023497	0.00017086	164272	314856
44	18	1.05653	0.441371	46240	87352	44	18	0.000268242	0.000170745	127312	330416
28	28	0.620279	0.63062	64576	48088	28	28	0.000209897	0.000207024	214840	135616
36	28	0.824124	0.671394	53760	49208	36	28	0.00023497	0.000209702	164272	145872
44	28	1.05653	0.673557	46240	54880	44	28	0.000268242	0.000210265	127312	163088
56	28	1.388	0.67004	39520	59568	56	28	0.000312829	0.000210444	100992	176344
72	28	1.79778	0.647573	34480	64152	72	28	0.000366755	0.000208479	80304	188928
44	44	1.05653	1.03645	46240	31152	44	44	0.000268242	0.000269072	127312	69952
56	44	1.388	1.11769	39520	34504	56	44	0.000312829	0.000278791	100992	78960
72	44	1.79778	1.14647	34480	38784	72	44	0.000366755	0.000274503	80304	91600
88	44	2.33344	1.13333	29840	42840	88	44	0.000436296	0.000269361	64464	103600
112	44	3.21416	1.12562	26240	46504	112	44	0.000526625	0.000267529	51088	112536

3.1 Distortion Models

In [6], the geometry distortion D_g and color distortion D_c are modeled as functions of the geometry and color quantization step sizes of the first frame ($Q_{g,1}$ and $Q_{c,1}$, respectively) according to

$$\begin{cases} D_g = \alpha_g Q_{g,1} + \delta_g \\ D_c = \alpha_c Q_{c,1} + \beta_c Q_{g,1} + \delta_c, \end{cases} \quad (3)$$

where $\alpha_g, \delta_g, \alpha_c, \beta_c$, and δ_c are model parameters. In this paper, we extend this model by including the quantization step sizes of all frames. For simplicity, we assume that the number of frames N is equal to 4. To study the effect of the quantization in the first frame on the distortion in the second frame, we fixed the quantization steps of the second frame and varied those of the first frame. Table 1 shows that the effect of the quantization step of the first frame on the distortion of the second frame is very small for both geometry and color. We observed the same phenomenon for the other frames. Consequently, we propose the following distortion models for the i th frame

$$\begin{cases} D_{g,i} = \alpha_{g,i} Q_{g,i} + \delta_{g,i} \\ D_{c,i} = \alpha_{c,i} Q_{c,i} + \beta_{c,i} Q_{g,i} + \delta_{c,i}, \end{cases} \quad (4)$$

where $\alpha_{g,i}, \delta_{g,i}, \alpha_{c,i}, \beta_{c,i}$, and $\delta_{c,i}$ are model parameters. The overall distortion is then modeled as

$$D = \frac{1}{4} (\sum_{i=1}^4 \omega D_{g,i} + (1 - \omega) D_{c,i}) \quad (5)$$

3.2 Rate Models

As the number of bits of the first frame is only determined by its own quantization steps ($Q_{g,1}, Q_{c,1}$), it can be modeled as in [6]

$$\begin{cases} R_{g,1} = \gamma_{g,1} Q_{g,1}^{\theta_{g,1}} \\ R_{c,1} = \gamma_{c,1} Q_{c,1}^{\theta_{c,1}} \end{cases} \quad (6)$$

where $\gamma_{g,1}, \gamma_{c,1}, \theta_{g,1}$, and $\theta_{c,1}$ are model parameters. To obtain the rate model for the second frame, we first ignore the impact of the first frame on the second frame and use the basic model

$$\begin{cases} R_{g,2} = \gamma_{g,2} Q_{g,2}^{\theta_{g,2}} \\ R_{c,2} = \gamma_{c,2} Q_{c,2}^{\theta_{c,2}} \end{cases} \quad (7)$$

where $\gamma_{g,2}, \gamma_{c,2}, \theta_{g,2}$, and $\theta_{c,2}$ are model parameters. However, Table 1 shows that the number of bits of the second frame increases when the quantization steps of the first frame increase. To take this dependency into account, we update the model as

$$\begin{cases} R_{g,2} = (\varphi_{g,(1,2)} \cdot Q_{g,1} + 1) \gamma_{g,2} Q_{g,2}^{\theta_{g,2}} \\ R_{c,2} = (\varphi_{c,(1,2)} \cdot Q_{c,1} + 1) \gamma_{c,2} Q_{c,2}^{\theta_{c,2}} \end{cases} \quad (8)$$

where $\varphi_{g,(1,2)}$ and $\varphi_{c,(1,2)}$ are the impact factors of the first frame on the second frame. Similarly, we first assume that the number of bits of the third and fourth frames are independent of the quantization steps of the other frames and model them as

$$\begin{cases} R_{g,3} = \gamma_{g,3} Q_{g,3}^{\theta_{g,3}} \\ R_{c,3} = \gamma_{c,3} Q_{c,3}^{\theta_{c,3}} \end{cases} \quad (9)$$

$$\begin{cases} R_{g,4} = \gamma_{g,4} Q_{g,4}^{\theta_{g,4}} \\ R_{c,4} = \gamma_{c,4} Q_{c,4}^{\theta_{c,4}} \end{cases} \quad (10)$$

where $\gamma_{g,3}, \gamma_{c,3}, \theta_{g,3}, \theta_{c,3}, \gamma_{g,4}, \gamma_{c,4}, \theta_{g,4}$, and $\theta_{c,4}$ are model parameters. Then we update the models as

$$\begin{cases} R_{g,3} = \prod_{i=1}^2 (\varphi_{g,(i,i+1)} \cdot Q_{g,i} + 1) \gamma_{g,3} Q_{g,3}^{\theta_{g,3}} \\ R_{c,3} = \prod_{i=1}^2 (\varphi_{c,(i,i+1)} \cdot Q_{c,i} + 1) \gamma_{c,3} Q_{c,3}^{\theta_{c,3}} \end{cases} \quad (11)$$

$$\begin{cases} R_{g,4} = \prod_{i=1}^3 (\varphi_{g,(i,i+1)} \cdot Q_{g,i} + 1) \gamma_{g,4} Q_{g,4}^{\theta_{g,4}} \\ R_{c,4} = \prod_{i=1}^3 (\varphi_{c,(i,i+1)} \cdot Q_{c,i} + 1) \gamma_{c,4} Q_{c,4}^{\theta_{c,4}} \end{cases} \quad (12)$$

where $\varphi_{g,(i,i+1)}$ and $\varphi_{c,(i,i+1)}$ ($i = 2, 3$) are the impact factors of the i -th frame on the $(i + 1)$ -th one. Finally, we use (6), (8), (11) and (12) to build the rate model as $R = \sum_{i=1}^4 R_{g,i} + R_{c,i}$.

Table 2. QP settings to determine the model parameters

Model parameters	$QP_{g,1}$	$QP_{g,2}$	$QP_{g,3}$	$QP_{g,4}$	$QP_{c,1}$	$QP_{c,2}$	$QP_{c,3}$	$QP_{c,4}$
$\alpha_{g,1}, \delta_{g,1}; \alpha_{g,2}, \delta_{g,2}; \alpha_{g,3}, \delta_{g,3}; \alpha_{g,4}, \delta_{g,4};$	30	30	30	30	40	40	40	40
$\alpha_{c,1}, \beta_{c,1}, \delta_{c,1}; \alpha_{c,2}, \beta_{c,2}, \delta_{c,2}; \alpha_{c,3}, \beta_{c,3},$	36	36	36	36	30	30	30	30
$\delta_{c,3}; \alpha_{c,4}, \beta_{c,4}, \delta_{c,4}$	38	38	38	38	28	28	28	28
	30	30	30	30	40	40	40	40
	36	36	36	36	30	30	30	30
	38	38	38	38	28	28	28	28
$\gamma_{g,1}, \theta_{g,1}; \gamma_{c,1}, \theta_{c,1};$	17	25	33	41	17	25	33	41
$\gamma_{g,2}, \theta_{g,2}; \gamma_{c,2}, \theta_{c,2};$	33	25	33	41	33	25	33	41
$\gamma_{g,3}, \theta_{g,3}; \gamma_{c,3}, \theta_{c,3};$	17	41	33	41	17	41	33	41
$\gamma_{g,4}, \theta_{g,4}; \gamma_{c,4}, \theta_{c,4}$	17	25	49	41	17	25	49	41
	19	24	29	34	19	24	29	34
	34	24	40	37	34	24	40	37
	27	41	37	45	27	41	37	45
	27	17	37	45	27	17	37	45

3.3 Model Parameters

To determine the parameters of the distortion models, we first encode the point cloud for three different sets of quantization steps ($\mathbf{Q}_g, \mathbf{Q}_c$) and compute the corresponding actual distortions and number of bits for each frame. Next, we solve the resulting system of equations to find $\alpha_{g,i}, \delta_{g,i}, \alpha_{c,i}, \beta_{c,i}, \delta_{c,i}$ ($i = 1, \dots, 4$). To determine the parameters of the rate models, we encode the point cloud for eight more sets of quantization steps and use linear regression in (7), (9), and (10) to estimate the parameters $\gamma_{g,i}, \theta_{g,i}, \gamma_{c,i}, \theta_{c,i}$ ($i = 1, \dots, 4$). Finally, the impact factors $\varphi_{g,(1,2)}, \varphi_{g,(2,3)}, \varphi_{g,(3,4)}, \varphi_{c,(1,2)}, \varphi_{c,(2,3)}$, and $\varphi_{c,(3,4)}$, are empirically set to

$$\begin{cases} \varphi_{g,(1,2)} = \varphi_{c,(1,2)} = 0.004 \\ \varphi_{g,(2,3)} = \varphi_{c,(2,3)} = 0.0015 \\ \varphi_{g,(3,4)} = \varphi_{c,(3,4)} = 0.0010. \end{cases} \quad (13)$$

Table 2 shows the QP settings used to compute the parameters of the distortion and rate models.

4 Optimization

To solve the rate-distortion optimization problem (2), we apply a DE variant to the analytical models derived in Section 3. Unlike the standard DE algorithm, this variant decreases the crossover rate with time and uses a random scaling factor. The decrease in crossover rate at runtime increases the exploitation pressure at the end of the run [12]. The randomization of the scaling factor is motivated by the experimental observation that a certain degree of randomization is beneficial [12].

The details of the implemented algorithm are as follows. A candidate solution (agent) for problem (2) is denoted by $\mathbf{x} = (\mathbf{Q}_g, \mathbf{Q}_c) = (x_1, x_2, \dots, x_{2N})$.

- Choose a population size NP , an interval I for the scaling factor, and a number of iterations n .
- Build a population of NP agents $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(NP)}$ such that each component $x_i^{(j)}, i = 1, \dots, 2N; j = 1, \dots, NP$, is randomly chosen in the set of quantization steps $\{q_0, \dots, q_{M-1}\}$ and $R(\mathbf{x}^{(j)}) \leq R_T$ for $j = 1, \dots, NP$.
- FOR $k = 1$ to n
 - If $k < \frac{2}{3}n$, set the crossover rate to $CR = 0.9$; otherwise, set $CR = 0.1$;
 - FOR $j = 1$ to NP
 - Step 1: Select randomly from the population three different agents $\mathbf{a}, \mathbf{b}, \mathbf{c}$ that are also different from $\mathbf{x}^{(j)}$
 - Step 2: Select randomly an index r such that $1 \leq r \leq 2N$
 - Step 3: Compute a candidate new agent $\mathbf{y}^{(j)}$ as follows:
 - For each $i \in \{1, \dots, 2N\}$, choose a random number r_i according to a uniform distribution in $(0,1)$. Choose a scaling factor w randomly in I .
 - If $r_i \leq CR$ or $i = r$, then set $y_i^{(j)} = a_i + w \times (b_i - c_i)$; otherwise, set $y_i^{(j)} = x_i^{(j)}$
 - If $y_i^{(j)} < q_0$, set $y_i^{(j)} = q_0$. If $y_i^{(j)} > q_{M-1}$, set $y_i^{(j)} = q_{M-1}$.
 - Step 4: If $D(\mathbf{y}^{(j)}) < D(\mathbf{x}^{(j)})$ and $R(\mathbf{y}^{(j)}) \leq R_T$, note j .
 - END FOR
 - FOR $j = 1$ to NP , replace $\mathbf{x}^{(j)}$ by $\mathbf{y}^{(j)}$ if j was noted in Step 4.
 - END FOR
 - END FOR
 - Select the agent from the population that gives the lowest distortion D and round the components of this agent to the nearest values in the set $\{q_0, \dots, q_{M-1}\}$.

Another way of solving problem (2) is to use conventional non-evolutionary constrained nonlinear optimization algorithms. However, when the problem is not convex, such algorithms are only guaranteed to find local minima and are very sensitive to the starting point of the algorithm (see Section 5).

Table 3. Accuracy of the proposed rate and distortion models.

Point cloud	Target bitrate	Model bitrate	Model distortion	Actual bitrate	Actual distortion	SCC		RMSE	
						Rate model	Distortion model	Rate model	Distortion model
<i>soldier</i>	65	65.21	30.52	63.21	30.59	0.9976	0.9984	21.16	0.33
	125	125.04	18.76	123.71	18.78				
	165	171.80	15.17	172.37	15.90				
	210	205.38	13.57	224.01	13.45				
	265	263.64	11.69	293.58	11.78				
	365	355.23	9.94	393.14	10.22				
<i>queen</i>	65	65.04	23.17	68.32	23.68	0.9984	0.9977	5.64	0.45
	125	124.85	16.82	129.83	17.40				
	165	171.66	14.89	170.03	15.52				
	210	207.36	14.00	204.68	14.41				
	265	265.68	13.00	272.96	13.15				
	365	356.95	12.07	366.55	12.09				
<i>loot</i>	65	66.87	12.72	65.85	13.18	0.9967	0.9989	24.72	0.24
	125	128.59	7.67	128.69	7.78				
	165	168.82	6.37	177.53	6.55				
	210	200.39	5.70	223.29	5.72				
	265	265.51	4.81	282.94	5.01				
	365	366.18	4.05	418.73	4.28				
<i>basket-ball-player</i>	30	30.20	12.34	27.72	12.07	0.9980	0.9988	6.27	0.13
	65	66.81	7.64	57.45	7.74				
	125	128.62	5.79	120.93	5.81				
	165	168.45	5.31	161.44	5.30				
	210	209.84	5.00	206.07	4.94				
	265	265.31	4.72	269.77	4.63				
<i>redand black</i>	90	88.74	19.47	84.38	19.81	0.9844	0.9974	52.28	0.55
	180	181.99	11.06	160.63	11.30				
	270	272.13	8.70	260.27	8.41				
	360	364.78	7.55	341.36	7.19				
	480	484.93	6.70	516.77	5.96				
	640	647.51	6.08	766.76	5.14				
<i>long-dress</i>	180	176.09	49.78	162.63	47.25	0.9980	0.9990	24.52	1.40
	270	268.93	38.66	251.10	37.42				
	360	364.74	33.00	350.09	31.96				
	480	489.00	28.85	487.61	28.03				
	640	639.86	25.83	681.42	24.62				
	840	850.50	23.33	884.63	22.60				
Average						0.9955	0.9984	22.43	0.51

5 Experimental Results

We first study the accuracy of the proposed distortion and rate models. The bitrates and distortions were computed for the quantization steps obtained as solutions of the optimization problem (2) for a given target bitrate. In the DE algorithm, the number of

iterations and the size of the population were set to 200 and 50, respectively. The interval I was $[0.1, 0.9]$. As in Section 3, we used the symmetric point-to-point distortions and considered only the luminance component. The weighting factor ω in (2) was set to 0.5. To compute the actual distortion and bit rates, we used TMC2 v12.0 [8] and encoded the first four frames of the point cloud for the IPPP GOP structure. Table 3 shows the results for six dynamic point clouds (longdress, redandblack, loot, soldier, queen, basketballplayer) [13,14]. The bitrates are expressed in kilobits per million points (kbpmp). We observe that the bitrates and distortions computed by our models have a high squared correlation coefficient (SCC) and a low root mean squared error (RMSE) with the actual values computed by encoding and decoding point clouds. This shows that our models are accurate.

Table 4. Bit allocation accuracy and BD [15] performance.

Point cloud	Target bitrate	[6]		BE	Proposed		BE	BD Distortion/ BD Color PSNR	BD bitrate/ BD Color bitrate
		Bitrate	Distortion		Bitrate	Distortion			
<i>soldier</i>	65	68.60	27.34	5.54%	63.21	30.59	2.75%		
	125	124.95	18.08	0.04%	126.51	19.12	1.21%		
	165	163.37	15.14	0.99%	174.36	15.56	5.67%	0.79/	21.68%/
	210	222.28	12.67	5.85%	213.68	13.69	1.75%	-0.20	9.46%
	265	296.15	10.91	11.75%	275.48	11.87	3.95%		
	365	414.56	9.51	13.58%	375.18	10.20	2.79%		
<i>queen</i>	65	59.87	24.36	7.90%	70.16	23.66	7.93%		
	125	121.09	17.02	3.13%	125.39	17.59	0.32%		
	165	162.14	15.29	1.73%	172.79	15.45	4.72%	0.85/	7.9%/
	210	204.88	14.18	2.44%	204.68	14.41	2.54%	-0.15	9.89%
	265	254.58	13.34	3.93%	267.73	13.22	1.03%		
	365	404.43	12.14	10.80%	366.50	12.09	0.41%		
<i>loot</i>	65	62.26	12.60	4.22%	65.26	12.95	0.41%		
	125	136.61	7.15	9.29%	129.43	7.81	3.54%		
	165	190.11	5.88	15.22%	177.57	6.53	7.62%	0.22/	9.11%
	210	195.26	5.78	7.02%	209.90	5.80	0.05%	-0.14	8.41%
	265	265.31	4.99	0.12%	283.20	5.02	6.87%		
	365	458.86	4.14	25.72%	409.71	4.32	12.25%		
<i>basketballplayer</i>	30	28.97	11.70	3.42%	27.72	12.07	7.61%		
	65	63.53	7.34	2.26%	60.72	7.50	6.58%		
	125	149.71	5.41	19.77%	122.02	5.78	2.38%	-0.10/	1.34%/
	165	198.11	5.03	20.07%	161.42	5.32	2.17%	0.04	-1.17
	210	276.95	4.62	31.88%	206.94	4.96	1.46%		
	265	376.71	4.39	42.15%	265.79	4.64	0.30%		
<i>redandblack</i>	90	84.86	18.99	5.71%	83.85	19.83	6.83%		
	180	157.75	11.17	12.36%	162.06	11.24	9.97%		
	270	269.82	8.08	0.07%	253.76	8.47	6.01%	0.02/	11.49%/
	360	348.28	6.94	3.25%	361.63	7.02	0.45%	-0.03	2.00%
	480	598.96	5.61	24.78%	520.08	5.93	8.35%		
	640	805.54	5.06	25.87%	737.76	5.19	15.28%		
<i>longdress</i>	180	167.65	47.05	6.86%	157.70	48.20	12.39%		
	270	307.27	33.70	13.81%	250.05	37.46	7.39%		
	360	424.25	29.52	17.85%	348.55	31.90	3.18%	-0.19/	-1.11%/
	480	597.98	26.07	24.58%	486.96	28.07	1.45%	0.03	-0.41
	640	784.40	24.15	22.56%	665.42	24.99	3.97%		
	840	1034.7	22.79	23.18%	890.33	22.66	5.99%		
Average				11.94%			4.65%	0.27/ -0.08	8.40%/ 4.70%

Table 4 compares the bit allocation accuracy of the proposed method to that of the method in [6]. The bit allocation accuracy is evaluated with the bitrate error (BE)

$$BE = \frac{|R_{actual} - R_{target}|}{R_{target}} \times 100\%, \quad (12)$$

where R_{actual} and R_{target} are the actual bitrate computed by the method and the target bitrate, respectively. The largest BE for the method in [6] was 42.15% (*basketball-player*, 265 kbpmp), while the largest BE for the proposed method was only 15.28% (*redandblack*, 640 kbpmp). Moreover, the average BE for the method in [6] was 11.94%, while that of the proposed method was only 4.65%. Table 4 and Fig. 2 show that the rate-distortion performance of the proposed method is slightly lower than that of the method in [6].

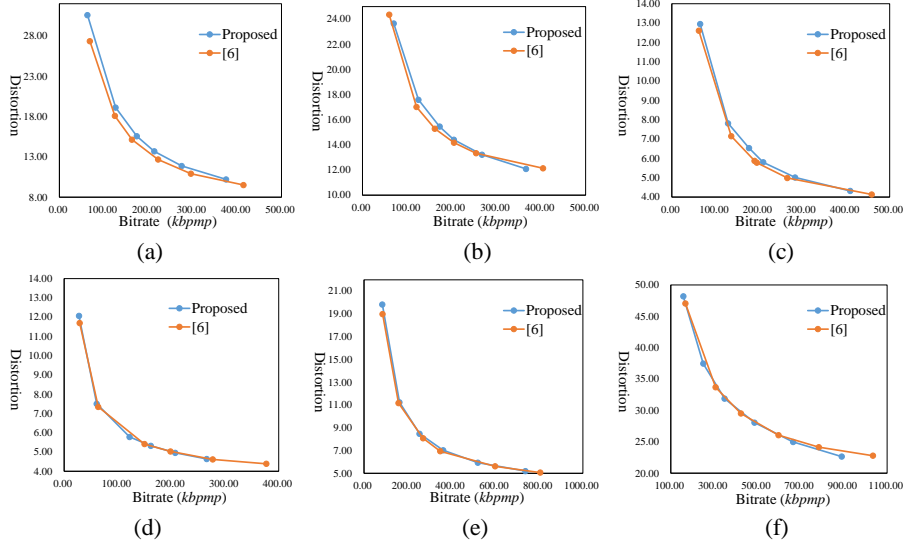


Fig. 2. Rate-distortion curves for the proposed DE-based method and the method in [6]. (a) *soldier*, (b) *queen*, (c) *loot*, (d) *basketballplayer*, (e) *redandblack*, (f) *longdress*.

Table 5 compares the time complexity of the proposed method to that of the method in [6]. The increase in the CPU time is mainly due to the pre-optimization step needed to determine the parameters of the models (11 encodings for the proposed method vs. three encodings for the method in [6]).

Table 5. CPU time on a laptop with a 2.7 GHz i7-7500U processor and 8 GB RAM.

Method	Pre-optimization (s)	Optimization (s)
[6]	3×3600	1.42
DE	11×3600	120
Full Search	0	$52^8 \times 3600$

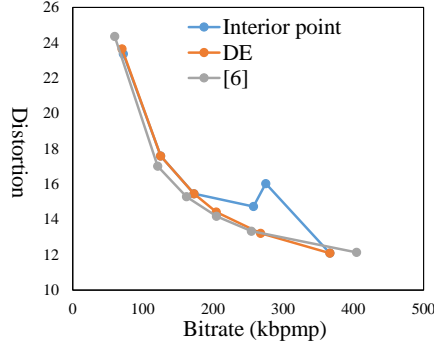


Fig. 3. Rate-distortion curves of DE, the method in [6], and the method in [16] for *queen*.

Finally, Fig. 3 illustrates how solving the optimization problem (2) with conventional non-evolutionary constrained nonlinear optimization algorithms can lead to poor solutions. Here the MATLAB implementation of the state-of-the-art interior point method in [16] was used with the starting point (2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5).

The data files used in the experimental results are available in [17].

6 Conclusion

We proposed analytical distortion and rate models for V-PCC that include the geometry and color quantization steps of all frames in a group of frames. Then, we used the models and a DE variant to efficiently select the quantization steps for a given target bitrate. Experimental results show that the proposed optimization technique allows a better rate control than the state-of-the-art. Rate control is critical in applications where the bandwidth is constrained. Our optimization technique can be easily extended to the case where the point cloud consists of more than one group of frames: we first determine the model parameters of the distortion and rate models for each group separately and then use DE to minimize the overall distortion subject to the constraint on the total number of bits. As further future work, we plan to apply our technique to GOPs of more than four frames and to the V-PCC random access configuration.

7 Acknowledgement

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 836192.

References

1. MPEG-3DG. V-PCC Codec Description. ISO/IEC JTC1/SC29/WG7 N00012, (2020).

2. MPEG-3DG. G-PCC Codec Description v9. ISO/IEC JTC1/SC29/WG7 N0011, (2020).
3. Sullivan G. J., Ohm J., Han W. and Wiegand T. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22(12) 1649-1668 (2012).
4. Price K., Storn R. M., and Lampinen J. A. *Differential Evolution: A Practical Approach to Global Optimization*. Springer. ISBN 978-3-540-20950-8, 2005.
5. Liu Q., Yuan H., Hou J., Liu H., and Hamzaoui R. Model-based encoding parameter optimization for 3D point cloud compression. In: *Proc. APSIPA Annual Summit and Conference*, pp.1981-1986, Honolulu, (2018).
6. Liu Q., Yuan H., Hou J., Hamzaoui R., and Su H. Model-based joint bit allocation between geometry and color for video-based 3D point cloud compression. In: *IEEE Transactions on Multimedia*, DOI: 10.1109/TMM.2020.3023294.
7. Liu Q., Yuan H., Hamzaoui R., and Su H. Coarse to fine rate control for region-based 3D point cloud compression. In: *Proc. IEEE ICME Workshops*, London, (2020).
8. V-PCC Test Model v12, ISO/IEC JTC 1/SC 29/WG 7 N00006.
9. Point Cloud Library (PCL), <http://pointclouds.org/>
10. Mekuria R., Li Z., Tulvan C., and Chou P. Evaluation criteria for pcc (point cloud compression). ISO/IEC JTC1/SC29/WG11, N16332, (2016).
11. HEVC test model, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware
12. Neri, F., Tirronen, V. Recent advances in differential evolution: a survey and experimental analysis. *Artif Intell Rev* 33, 61–106 (2010).
13. MPEG point cloud datasetscfp, <http://mpegfs.intevry.fr/MPEG/PCC/DataSets/pointCloud/CfP/datasets>
14. MPEG point cloud datasets-AnimatedPCMicrosoft, <http://mpegfs.intevry.fr/MPEG/PCC/DataSets/AnimatedPC-Microsoft>.
15. Bjøntegaard G. Calculation of average PSNR differences between RD-curves, <http://wftp3.itu.int/av-arch/videosite/0104-Aus/VCEG-M33.doc>.
16. Byrd R.H., Hribar M. E., and Nocedal J. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization* 9(4), 877–900 (2006).
17. Yuan, H., Hamzaoui, R., Neri, F., and Yang, S. (2021, June 26). Data files for ICIG 2021 paper (Version 1.0). Zenodo. <http://doi.org/10.5281/zenodo.5034575>.